ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

vk.com/id446425943
vk.com/club152685050

ассистент		А. А. Фоменкова
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №7

ОРГАНИЗАЦИЯ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ ПРИ РЕШЕНИИ ЧИСЛЕННЫХ ЗАДАЧ

по курсу: ИНФОРМАТИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4736		
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2017

Цель работы

Научиться организовывать текстовый интерфейс пользователя. Получение навыков написания программ для решения численных задач с текстовым интерфейсом пользователя

Задание на лабораторную работу

Проводятся измерения угла крена самолета с помощью гироскопического датчика угловых скоростей. Необходимо написать программу в MATLAB, отображающую изменение угла крена во времени. Считать, что на выходе датчиков имеется аналоговый сигнал, который аппроксимируется аналитически заданной функцией. В начальный момент времени значение угла крена равнялось 0.

Программа должна работать следующим образом:

1. При запуске программы консоль и рабочая область MATLAB должны быть очищены. В консоль должна быть выведена короткая информация о программе (что реализует программа) и главное меню программы, содержащее пункты согласно номеру варианта (таблица 7.5)

2. Программа должна запросить у пользователя пункт меню. Пункт меню, введенный пользователем, должен быть проверен на корректность ввода. Если пункт меню введен некорректно, на экран должно быть выведено сообщение об ошибке и пользователю должно быть предложено заново ввести пункт меню.

3. Для вариантов б, д – при выборе 2 пункта меню из основной программы должна быть вызвана подпрограмма-1, которая выведет в консоль подменю-1 согласно номеру варианта (пункт с * - дополнительное задание).

4. При выборе пользователем пункта меню «Расчет угла крена самолета» программа должна запросить у пользователя:

vk.com/id446425943
vk.com/club152685050

- Функцию угловой скорости (как строку),
- Время в минутах, в течение которого производились измерения. Введенное пользователем значение должно быть проверено на корректность ввода (положительное число). Количество точек дискретизации исходной функции определить самостоятельно.

Для расчета угла крена по значению угловой скорости крена использовать интегрирование угловых скоростей методом прямоугольников (см. ЛР №3). Вывод результатов расчета осуществить аналогично демонстрационному примеру (см.п.п. 5 задания на лабораторную работу №6). После расчета значения угла крена по данным пользователя, программа должна вывести подменю-2 (пункт со * - дополнительное задание).

Дополнительное задание ***

В п.п.4 запросить функцию угловой скорости не как строку, а как вектор значений (дискретный сигнал). Введенные пользователем данные проверить на корректность ввода (введен числовой вектор).

5. При выборе пункта меню «Демонстрация работы программы» в консоль должны быть выведены результаты с соответствующими комментариями: функция угловой скорости крена и таблица с результатами расчета угла крена (согласно номеру варианта, таблица 7.4). Данные для демонстрационного примера выбрать в соответствии с номером варианта (таблица 7.5). После вывода данных программа должна вывести сообщение «Для возврата в главное меню нажмите любую клавишу».

6. При выборе пункта меню «Информация о программе и авторе» в консоль должны быть выведены сведения: Название программы, ее назначение, используемый метод решения, дата последней редакции, ФИО, группу разработчика, контакты. После вывода информации программа должна вывести сообщение «Для возврата в главное меню нажмите любую клавишу».

vk.com/id446425943
vk.com/club152685050

Таблица 7.4 Варианты заданий

Вариант	Главное меню	Подменю-1	Подменю-2	Вывод результатов
в	1. Демонстрация работы программы 2. Расчет угла крена 3. Информация о программе и авторе 4. Выход		1. Задать новые значения угловой скорости* 2. Возврат в главное меню 3. Завершение работы программы	Расчеты вывести в таблице с 2 столбцами – время, угол.

Таблица 7.5

№ варианта	Задание	Функция для демонстрационного примера
13	в	

Ход работы

1. Главная подпрограмма(Главный цикл)

1.1 Листинг

```
%Очистка
clear all;
clc;

userStr = '';
time = -1;
vecX = []; vecY = []; vecZ = [];

inform();
menuState = 'M';
nextMenuState = 'M';

while (true)

	switch menuState
		case 'M' %Главное меню
			disp('Главное меню AngleOfRoll');
			disp('1 - Демонстрация работы программы');
			disp('2 - Расчет угла крена');
			disp('3 - Информация о программе и авторе');
			disp('4 - Выход');
			menuState = inputMenu(4);
			clc;
			continue;
		case 'S' %Подменю 2
			disp('Меню расчетов');
			disp('1 - Задать новые значения');
			disp('2 - Возврат в главное меню');
			disp('3 - Завершение работы');
			menuState = inputMenu(3);
			menuState = sprintf('S%s', menuState);
			clc;
			continue;
		case '1' %Демонстрация
			disp('Демонстрация программы');
			[vecXT, vecYT, vecZT] = countAngleOfTime('0.05*sin(t^2)', 10);
			printTable2D(vecXT, vecYT);
			disp('Для возврата в главное меню нажмите любую клавишу')
			pause;
			clc;
			nextMenuState = 'M';
		case '2' %Расчет
			if(isequal(userStr, '') || time < -1)
				menuState = 'I';%Принудительно назначает стадию ввода
				nextMenuState = '2';
				continue;
			end
			printTable2D(vecX, vecY);
			%Расчет успешен.
			menuState = 'S';
			nextMenuState = '2';
			continue;
		case '3' %Информация
			inform();
			nextMenuState = 'M';
		case '4' %Выход
			clc;
```

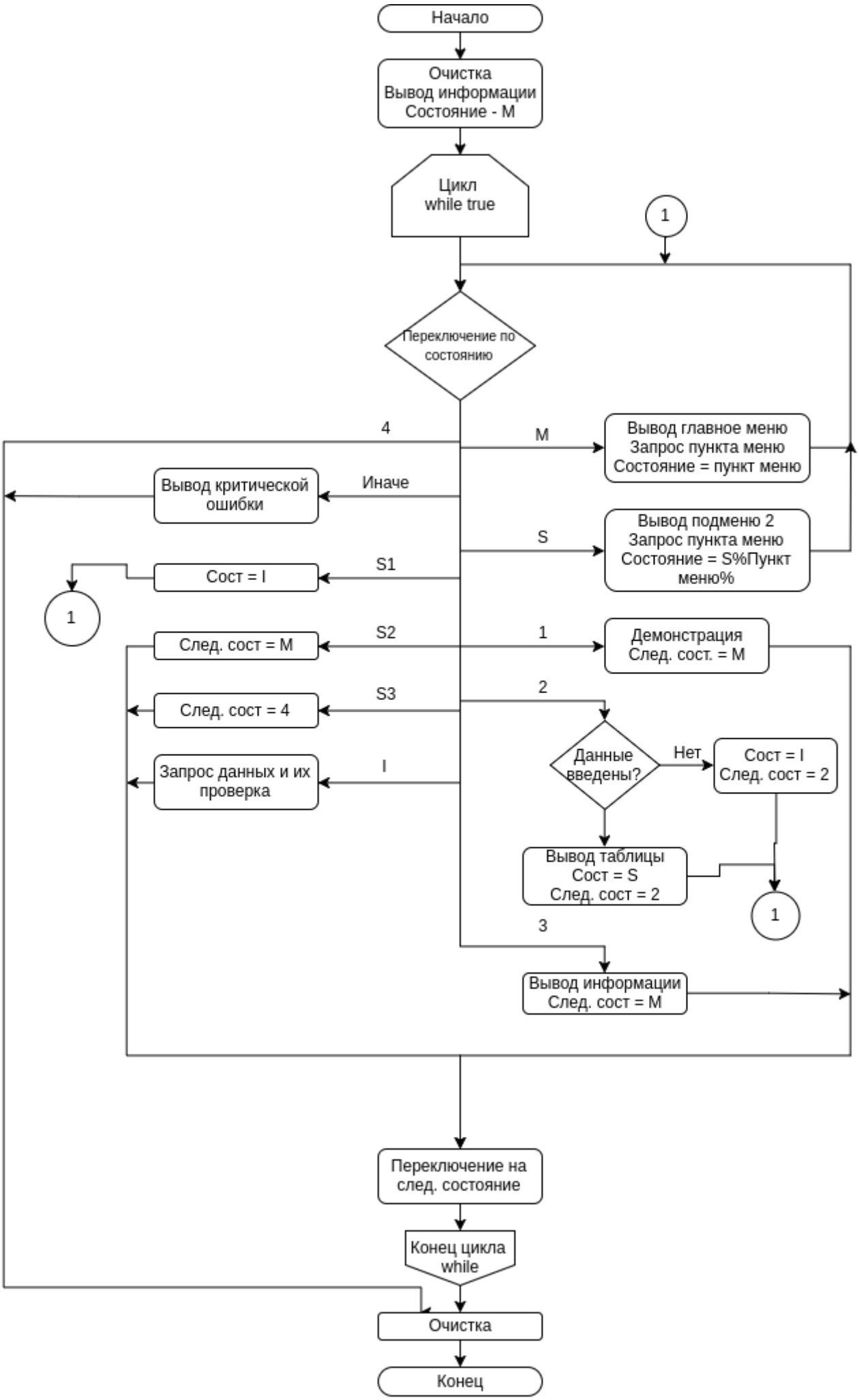
vk.com/id446425943
vk.com/club152685050

```
                clear all;
                close();
                break;
        case 'S1' %Ретранслятор с подменю 2 до меню выбора
                menuState = 'I';
                continue;
        case 'S2' %Ретранслятор с подменю 2 до главного меню
                nextMenuState = 'M';
        case 'S3' %Ретранслятор с подменю 2 до меню выхода
                nextMenuState = '4';
        case 'I' % Запрашивает данные
                while(true)
                        userStr = input('Введите f(t) или вектор дискертных значений: ', 's');
                        userStr = strrep(userStr, ',', '.');
                        if(~isempty(str2num(userStr)))
                                sizeMatrix = size(str2num(userStr));
                                if(sizeMatrix(1) > 1)
                                        disp('Произошла ошибка ввода, повторите попытку.');
                                        continue;
                                end
                                disp('Введите интервал времени между измерениями в векторе.');
                                break;
                        else
                                disp('Введите время измерения.');
                                break;
                        end
                end
                time = inputTime();
                disp('Начинается предварительный расчет,
                ожидайте.');[vecX, vecY, vecZ] = count4736(userStr,
                time); clc;
        otherwise
                clc;
                clear all;
                close();
                disp('Критическая ошибка. Доложите об этом!');
                break;
    end

    menuState = nextMenuState;

end
clear all;
```

1.2 Блок схема


Начало
Очистка
Вывод информации
Состояние - M
Цикл
while true
1
Переключение по состоянию
4
M
Вывод главное меню
Запрос пункта меню
Состояние = пункт меню
Иначе
Вывод критической ошибки
S
Вывод подменю 2
Запрос пункта меню
Состояние = S%Пункт меню%
S1
Сост = I
1
S2
1
Демонстрация
След. сост. = M
След. сост = M
S3
2
След. сост = 4
Данные введены?
Нет
Сост = I
След. сост = 2
I
Запрос данных и их проверка
Вывод таблицы
Сост = S
След. сост = 2
1
3
Вывод информации
След. сост = M
Переключение на след. состояние
Конец цикла
while
Очистка
Конец

1.3 Вывод программы

Группа: vk com club152685050
1 - Демонстрация работы программы
2 - Расчет угла крена
3 - Информация о программе и авторе
4 - Выход
Введите пункт меню: 4
>>

2. Подпрограмма countAngleOfTime.m

2.1 Листинг

```
function [vecX vecY vecZ] = countAngleOfTime(inputStr, time)
%countAngleOfTime - Считает угол.

    Syntax:[vecX vecY vecZ] = countAngleOfTime(inputStr, time)
% vecX - вектор значения времени t
% vecY - вектор значения угла
% vecZ - вектора значения угловой скорости
% Если во входной строке вектор, то time - время между измерениями вектора
% Если во входной строке f(t), то time время, до какого момента считается
% интеграл
```

vk.com/id446425943
vk.com/club152685050

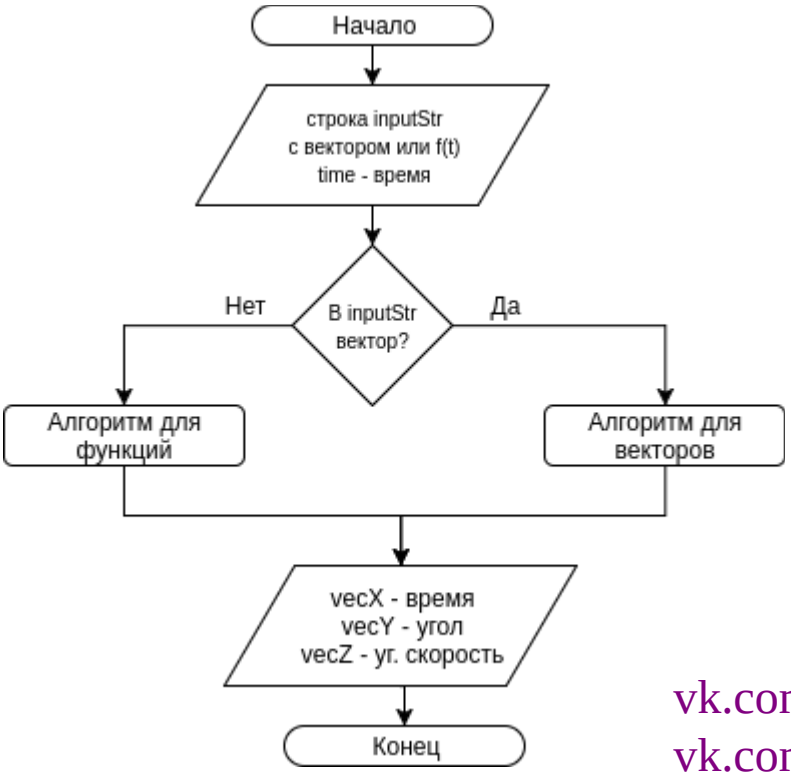
```
    if ( isempty( str2num(inputStr) ) )
        %В строке не вектор, используем алгоритм для функции
        iMax = time/0.1 + 1;
        vecX = 0:0.1:time;
        vecY(1:iMax) = 0;
        vecZ(1:iMax) = 0;

        for i = 1:iMax
            %Расчитываем угловую скорость
            t = vecX(i);
            vecZ(i) = eval(inputStr);
            %Расчитываем угол
            y = 0;
            x = vecX(i);
            for t = 0:0.01:x
                y = y + eval(inputStr);
            end
            vecY(i) = y*0.01;
        end
        return;
    else
        % Алгоритм для векторов
        vector = str2num(inputStr);

        iMax = numel(vector) + 1;
        vecX = 0:time:((iMax-1)*time);
        vecY(1:iMax) = 0;
        vecZ(2:iMax) = vector;
        for i = 1:iMax
            vecY(i) = sumVector(vecZ, 1, i) * time; % Это интеграл
        end
        return;
    end
```

```
end
```

2.2 Блок схема



vk.com/id446425943
vk.com/club152685050

2.3 Вывод программы

```
>> [a b c] = countAngleOfTime('sin(t)', 1)
a =
         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000    0.7000    0.8000    0.9000
1.0000
b =
         0    0.0055    0.0209    0.0461    0.0809    0.1248    0.1775    0.2384    0.3069    0.3823
0.4639
c =
         0    0.0998    0.1987    0.2955    0.3894    0.4794    0.5646    0.6442    0.7174    0.7833
0.8415
```

3. подпрограмма *inform* выводит информацию о программе

3.1 Листинг

```
function [] = inform()
%inform - Выводит информацию
%
% Syntax: inform()

disp('Программа 4736');


end
```

3.2 Вывод

```
>> inform
Программа AngleOfRoll
```

4. Функция *inputMenu* запрашивает номер меню

4.1 Листинг

```
function [menu] = inputMenu(menus)
%inputMenu - Запрашивает целочисленое число от 1 до input



%
% Syntax: [menu] = inputMenu(menus)

	while (true)

		strInput = input('Введите пункт меню: ', 's');
		strInput = strrep(strInput, ',', '.'); %Замена запятых на точки для корректной работы с
десятичными дробями
		in = str2double(strInput);
		if(isnan(in))
			disp('Ошибка ввода.');
			continue;
		end
		onlyFractPart = in - floor(in);
		if(onlyFractPart == 0) % Проверка на целочисленость
			if(in >= 1 && in <= menus)
				menu = strInput;
				return;
			end
		end
		disp('Ошибка ввода.');
	end
end
```

4.2 Вывод

```
>> inputMenu(3)
Введите пункт меню: 5
Ошибка ввода.
Введите пункт меню: S
Ошибка ввода.
Введите пункт меню: 2
ans =
   '2'
```

vk.com/id446425943
vk.com/club152685050

5. Функция inputTime запрашивает положительное число

5.1 Листинг

```
function [out] = inputTime()
%inputTime - Запрашивает положительное число у пользователя
%
% Syntax: [out] = inputTime()

	while(true) %Цикл
		in = input('Введите положительное число: ', 's');
		in = strrep(in, ',', '.');
		digit = str2double(in);
		if (isnan(digit) || digit < 0) %Если не число запрашиваем опять
			disp('Неверный ввод. Повторите попытку');
			continue;
		else
			out = digit;
		return;
		end
```

```
        end
end
```

5.2 Вывод

vk.com/id446425943
vk.com/club152685050

```
>> inputTime()
Введите положительное число: s
Неверный ввод. Повторите попытку
Введите положительное число: -3
Неверный ввод. Повторите попытку
Введите положительное число: 9.9999
ans =
   9.9999
```

6. *sumVector* складывает элементы вектора

6.1 Листинг

```
function [S] = sumVector(vector, ibegin, iend)
%sumVector - Складывает все элементы вектора от ibegin до iend
%
% Syntax: S = sumVector(vector, ibegin, iend)

S = 0;

for i = ibegin:iend
        S = S + vector(i);
end
```

6.2 Вывод

```
>> vec = [90 -90 30 -30 9 0 16 32 -0];
>> sumVector(vec, 3, 6)
ans =
    9
```

Вывод

С помощью функций ввода и вывода можно организовать текстовый интерфейс пользователя это открывает множество возможностей по работе с данными определенные пользователем, а не программистом.